

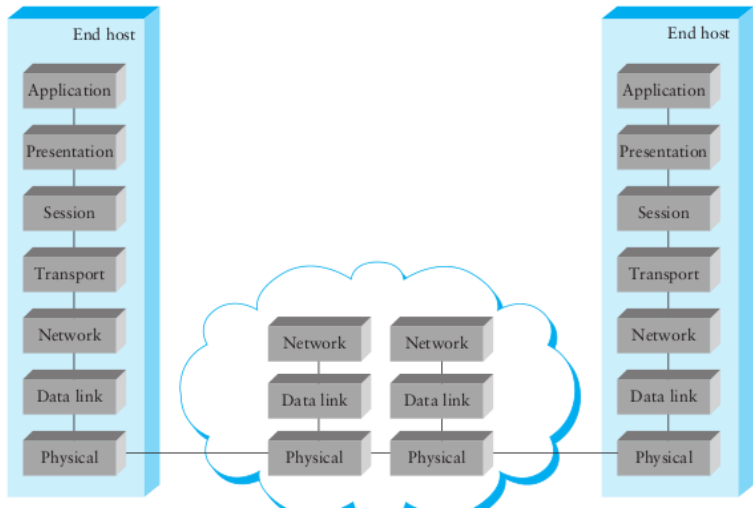
# Nivel de enlace

## Teoría de la Comunicaciones

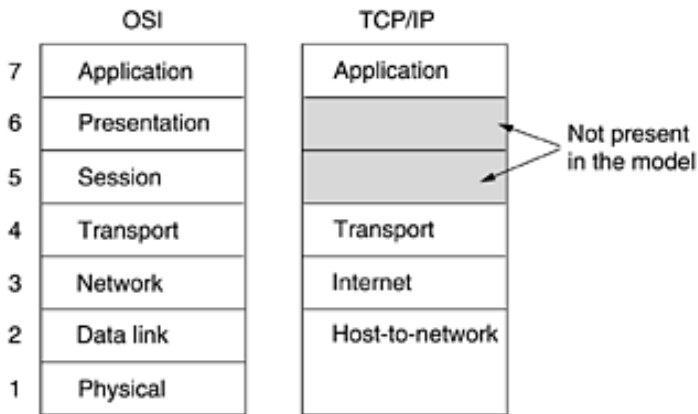
20 de Agosto 2014

# Arquitectura en capas

Las comunicaciones se dan en capas que se brindan servicios entre sí

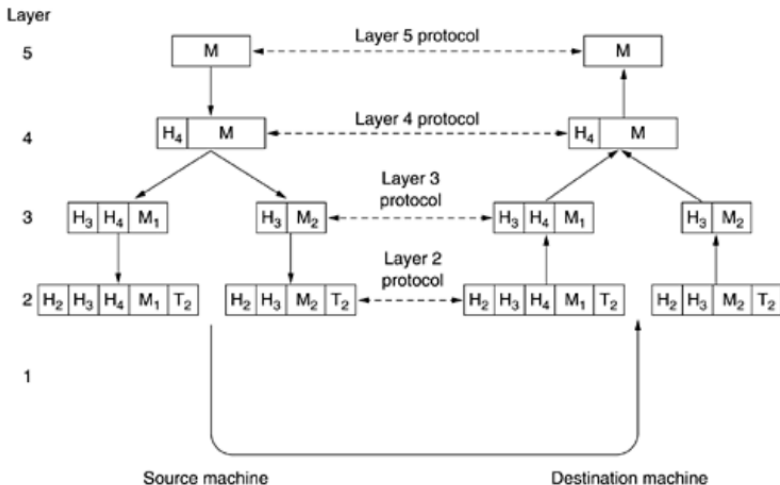


## OSI-ISO vs INTERNET



# Encapsulamiento

Cada capa implica el agregado de información de control en forma de *encabezados*



# Nivel de Enlace

## Conceptos

- **Caño serial** (no hay desorden)
- **Sujeto a ruido impulsivo** Lo que se recibe puede no ser lo que se envió (*error de transmisión*)

## Objetivos

- **Framing** - Encapsular los bits en frames agregando información de control
- **Proveer servicio a la capa superior**

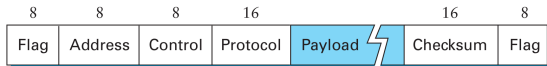
*¿Confiable o no confiable?*

- **Control de Errores** - ¿Se produjo algún error? ¿Qué hacemos con los errores?
- **Control de Flujo** - (Más adelante: en nivel de transporte)

¿Cómo se separan los frames en un tren de bits?

- Largo fijo
- Largo en el encabezado
- Delimitadores con bit-stuffing
- Violación de código

Ejemplo PPP:



- Sin conexión y sin reconocimiento
  - ★ Los datos se envían sin necesidad de saber si llegan bien.
- Sin conexión y con reconocimiento
  - ★ Los datos se envían y se asegura la correcta recepción mediante el aviso explícito (ACKs).
- Orientado a conexión
  - ★ Además de asegurar la correcta recepción de los datos. Se mantiene un *estado* de conexión (una sesión).

- Detección y Corrección de errores

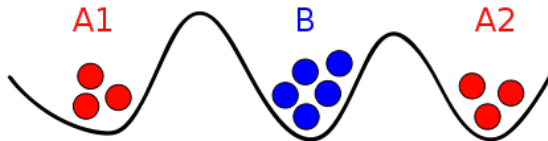
- ★  $m$  bits (**datos**) +  $r$  bits (**redundancia**) =  $n$  bits (**codeword**)
- ★ Sea  $d$  la mínima distancia de Hamming entre las codewords.
- ★ Sea  $e$  la cantidad de bits erróneos en la transmisión.
- ★  $\Rightarrow e + 1 \leq d$  para detectar y  $2e + 1 \leq d$  para corregir.

- Retransmisiones

- ★ Explícitas (mensajes de control específicos para pedir un dato nuevamente)
- ★ Implícitas (cuando ocurre un time-out se asume que el dato se perdió)



## El problema de los dos generales



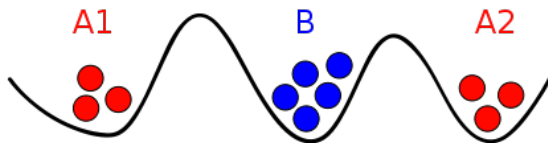
A1 a A2: "Atacaremos el 4 de agosto a las 09:00"

A2 a A1: "Recibido: Atacaremos 4 de agosto a las 09:00"

A1 a A2: "Recibido: Recibido: Atacaremos 4 de agosto a las 09:00"

...

## El problema de los dos generales



A1 a A2: "Atacaremos el 4 de agosto a las 09:00"

A2 a A1: "Recibido: Atacaremos 4 de agosto a las 09:00"

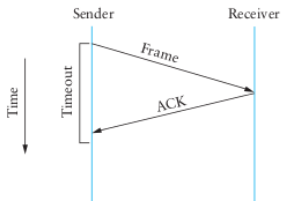
A1 a A2: "Recibido: Recibido: Atacaremos 4 de agosto a las 09:00"

...

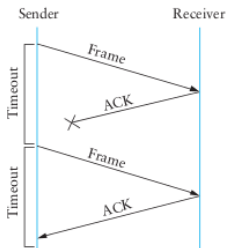
No existe un algoritmo para la confiabilidad

★ Enviamos solo un reconocimiento (ACK)

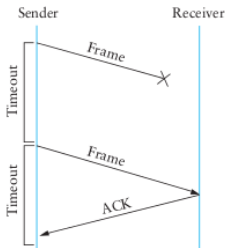
# Transmisión confiable: Stop and Wait



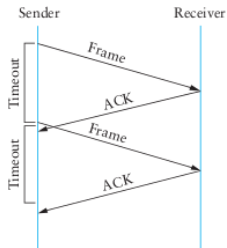
(a)



(c)



(b)



(d)

- Cada Frame debe ser reconocido por el receptor.
- Surge la necesidad de secuenciar los frames para evitar el problema de las reencarnaciones.
- ★ Para Stop & Wait, se necesitan por lo menos secuenciar 2 frames.

## Ejercicio

Un protocolo sobre un enlace punto a punto de 1Mbps y 0.25 segundos de delay, trabaja con stop & wait usando frames de largo fijo 2Kb y un CRC de 16bits para detectar errores.

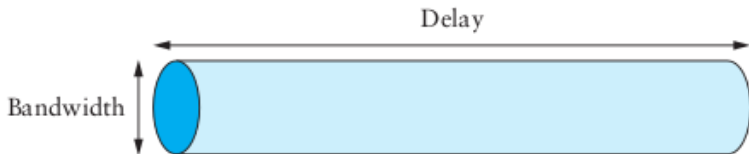
- a. Calcule cuánto tiempo es necesario para transmitir 20Mb de datos asumiendo que no hay errores.
- b. Ídem para un enlace con el mismo delay y 1 Gbps.
- c. Ídem para un enlace con la misma velocidad de transmisión y 0.1 segundos de delay.

¿Cuánto tiempo se está transmitiendo con respecto al tiempo bloqueado esperando?

$$Eficiencia = \frac{T_{tx}}{RTT}$$

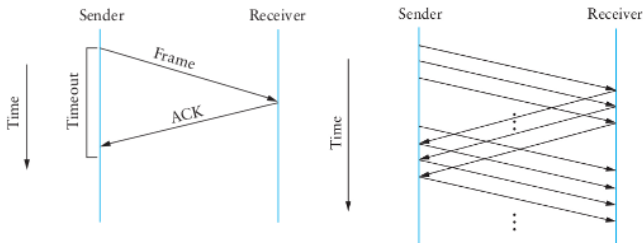
★ Aumentar la eficiencia es estar bloqueado lo menos posible.

# Capacidad de un canal: Producto $Bandwidth \times Delay$



- Multiplicando la velocidad de transmisión por el delay se obtiene la cantidad de bits que entran en un canal
- ★ Para aprovechar mejor, deberíamos calcular cuantos entran en el canal hasta que llega el primer ACK

## Motivación: Llenar el canal

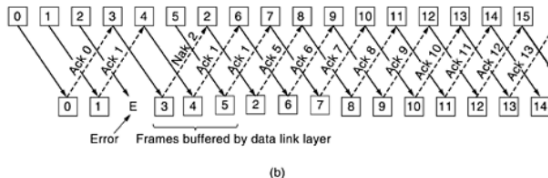
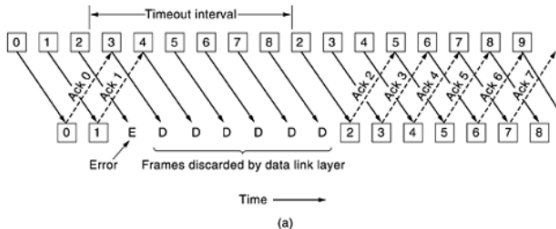


★ **Ventana de emisión:**  $SWS = V_{tx} * RTT / |Frame|$

★ **Enviar según:**

$$UltimoFrameEnviado \leq UltimoFrameReconocido + SWS$$

# Transmisión confiable: Selective ACKs



★ **Ventana de recepción:**  $RWS = \begin{cases} SWS & \text{Si hay SACK} \\ 1 & \text{Si no} \end{cases}$



Debemos aumentar la ventana de emisión para aprovechar mejor el canal:

$$\star SWS = V_{tx} * RTT / |Frame|$$

El receptor puede bufferear o no dependiendo del esquema de ACKs

$$\star RWS = \begin{cases} SWS & \text{Si hay SACK} \\ 1 & \text{Si no} \end{cases}$$

Y para distinguir reencarnaciones

$$\star \#frames \geq SWS + RWS$$

## Ejercicio

Un protocolo para un enlace pto-ptto de 100 Mbps y 2 segundos de delay trabaja con ventana deslizante, sin ACK Selectivo y opera con un frame de largo fijo de 800 bits que está compuesto por los siguientes campos (Asumir CRC de 16bits):

#SEQ ; #ACK ; Datos ; CRC

- a. ¿Cuántos bits deberían usarse para los campos de #SEQ y #ACK de manera de aprovechar lo mejor posible el canal?
- b. Calcule el rendimiento del frame de datos.